

ROSALES, Angel Abegail B.

BSIT 3-5

Activity 6: Using APPLY Family FUNCTIONS

You are given a dataset with students' grades in three subjects: Math, Science, and English.

```
students <- data.frame(  
  Name = c("John", "Alice", "Bob", "David", "Emma"),  
  Math = c(85, 90, 78, 92, 88),  
  Science = c(82, 88, 76, 95, 85),  
  English = c(89, 91, 80, 88, 94)  
)
```

1. Using the `apply()` function, calculate the **average score** for each student across all three subjects (Math, Science, and English). Add a new column to the students dataset that contains the average score.

```
> students <- data.frame(  
+   Name = c("John", "Alice", "Bob", "David", "Emma"),  
+   Math = c(85, 90, 78, 92, 88),  
+   Science = c(82, 88, 76, 95, 85),  
+   English = c(89, 91, 80, 88, 94)  
+ )  
>  
> students$Average <- apply(students[, c("Math", "Science", "English")], 1, mean)  
> print(students)  
  Name Math Science English Average  
1 John   85      82     89 85.33333  
2 Alice  90      88     91 89.66667  
3 Bob   78      76     80 78.00000  
4 David 92      95     88 91.66667  
5 Emma  88      85     94 89.00000
```

2. Using the `lapply()` function, calculate the **mean score** for each subject (Math, Science, English). Display the result as a list where each element represents the mean score of a subject.

```
> students <- data.frame(  
+   Name = c("John", "Alice", "Bob", "David", "Emma"),  
+   Math = c(85, 90, 78, 92, 88),  
+   Science = c(82, 88, 76, 95, 85),  
+   English = c(89, 91, 80, 88, 94)  
+ )  
> subject_means <- lapply(students[, c("Math", "Science", "English")], mean)  
> print(subject_means)  
$Math  
[1] 86.6  
  
$Science  
[1] 85.2  
  
$English  
[1] 88.4
```

3. Using the `apply()` function, calculate the **total score** for each subject (Math, Science, and English). Display the result as a vector where each element represents the total score of a subject.

```
> students <- data.frame(
+   Name = c("John", "Alice", "Bob", "David", "Emma"),
+   Math = c(85, 90, 78, 92, 88),
+   Science = c(82, 88, 76, 95, 85),
+   English = c(89, 91, 80, 88, 94)
+ )
>
> total <- sapply(students[, c("Math", "Science", "English")], sum)
>
> # Print the result
> print(total)
      Math Science English
      433     426     442
```

4. Using the `apply()` function, calculate the **highest score** in each subject (Math, Science, and English). Use `apply()` to perform the operation on the columns (i.e., find the maximum value in each column).

```
> students <- data.frame(
+   Name = c("John", "Alice", "Bob", "David", "Emma"),
+   Math = c(85, 90, 78, 92, 88),
+   Science = c(82, 88, 76, 95, 85),
+   English = c(89, 91, 80, 88, 94)
+ )
> highest <- apply(students[, c("Math", "Science", "English")], 2, max)
>
> # Print the result
> print(highest)
      Math Science English
      92     95     94
```

5. Using `lapply()`, create a new list where each element contains a vector of **scores greater than or equal to 85** for each subject. For example, in the "Math" subject, the result should be a list of scores in that subject that are greater than or equal to 85.

```
> students <- data.frame(
+   Name = c("John", "Alice", "Bob", "David", "Emma"),
+   Math = c(85, 90, 78, 92, 88),
+   Science = c(82, 88, 76, 95, 85),
+   English = c(89, 91, 80, 88, 94)
+ )
> scores <- lapply(students[, c("Math", "Science", "English")], function(x) x[x >= 85])
> print(scores)
$Math
[1] 85 90 92 88

$Science
[1] 88 95 85

$English
[1] 89 91 88 94
```

6. Using `apply()`, calculate the **standard deviation** of scores for each subject. Display the result as a vector where each element is the standard deviation of a subject.

```
> students <- data.frame(  
+   Name = c("John", "Alice", "Bob", "David", "Emma"),  
+   Math = c(85, 90, 78, 92, 88),  
+   Science = c(82, 88, 76, 95, 85),  
+   English = c(89, 91, 80, 88, 94)  
+ )  
>  
> standard_dev <- sapply(students[, c("Math", "Science", "English")], sd)  
> print(standard_dev)  
      Math Science English  
5.458938 7.049823 5.224940
```

7. Using the `apply()` function, create a new column in the students dataset that contains "Pass" if the average score of the student is greater than or equal to 85, and "Fail" if it is less than 85. Use `apply()` to check each student's average score and assign the respective result.

```
> students <- data.frame(  
+   Name = c("John", "Alice", "Bob", "David", "Emma"),  
+   Math = c(85, 90, 78, 92, 88),  
+   Science = c(82, 88, 76, 95, 85),  
+   English = c(89, 91, 80, 88, 94)  
+ )  
> students$Average <- apply(students[, c("Math", "Science", "English")], 1, mean)  
> students$Result <- ifelse(students$Average >= 85, "PASS", "FAIL")  
> print(students)  
  Name Math Science English Average Result  
1 John   85      82      89 85.33333 PASS  
2 Alice  90      88      91 89.66667 PASS  
3 Bob    78      76      80 78.00000 FAIL  
4 David  92      95      88 91.66667 PASS  
5 Emma   88      85      94 89.00000 PASS
```

8. Create a new list using `lapply()` that contains the **count of scores greater than or equal to 90** for each subject. Then, use the `apply()` function to find the total number of students with scores greater than or equal to 90 across all subjects.

```
> students <- data.frame(
+   Name = c("John", "Alice", "Bob", "David", "Emma"),
+   Math = c(85, 90, 78, 92, 88),
+   Science = c(82, 88, 76, 95, 85),
+   English = c(89, 91, 80, 88, 94)
+ )
>
> count <- lapply(students[, c("Math", "Science", "English")], function(x) sum(x >= 90))
> print(count_gte_90)
$Math
[1] 2

$Science
[1] 1

$English
[1] 2

>
> students_with_90 <- apply(students[, c("Math", "Science", "English")], 1, function(row) any(row >= 90))
> total_students <- sum(students_with_90)
>
> print(total_students)
[1] 3
>
```